

An evolutionary approach to the identification of Cellular Automata based on partial observations

Witold Bołt^{†*}, Jan M. Baetens^{*} and Bernard De Baets^{*}

[†]Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

^{*}KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Ghent, Belgium

Abstract—In this paper we consider the identification problem of Cellular Automata (CAs). The problem is defined and solved in the context of partial observations with time gaps of unknown length, *i.e.* pre-recorded, partial configurations of the system at certain, unknown time steps. A solution method based on a modified variant of a Genetic Algorithm (GA) is proposed and illustrated with brief experimental results.

I. INTRODUCTION

CAs present an attractive and effective modelling technique for a variety of problems. In order to use CAs in a practical modelling task, one needs to understand the underlying rules, relevant to the given phenomenon, and translate them into a CA local rule. Additionally, the state space, tessellation and neighborhood structure need to be pinned down beforehand. This narrows the application area for CAs, since there are problems for which it is hard to manually design a proper local rule. In some cases only the initial and final states of the system are known (*e.g.* [1]–[3]). Such problems motivate the research on automated CA identification. Various methods have been used, including genetic algorithms (GAs) [4]–[7], genetic programming [8]–[10], gene expression programming [11], ant intelligence [12], machine learning [13], as well as direct search/construction approaches [14]–[17].

Existing methods can be divided into two main groups. Firstly, methods for solving specific, global problems. An example of such a problem is majority classification in which one only knows the initial condition and the desired outcome. Secondly, methods that exploit the entire time series of configurations, where it is assumed that all configurations are known. Only limited research efforts have been devoted to problems involving identification based on partial information [4].

The main goal of the research presented in this paper is to develop methods capable of automated CA identification in case of partial information. The paper is organized as follows. In Section II we start with introducing basic definitions and presenting some well-known facts on CAs. Section III holds the formal definition of the CA identification problem, while in Section IV we reformulate this problem as an optimization task. In Section V the evolutionary algorithm for solving the identification problem is presented. The paper is concluded by Section VI which presents initial results of computational experiments.

An introduction to the methods presented in this paper, and a simpler formulation of the discussed algorithm can be found in [18].

II. PRELIMINARIES

We start by defining a CA. In this paper we will concentrate on 1D, deterministic CAs with a symmetric neighborhood.

Let $r \in \mathbb{N}$ and $f_A : \{0, 1\}^{2r+1} \rightarrow \{0, 1\}$ be any function, then for $N > 0$ we define the N -cell global CA rule $A_N : \{0, 1\}^N \rightarrow \{0, 1\}^N$ as:

$$A_N(\dots, s_i, \dots) = (\dots, f_A(s_{i-r}, \dots, s_{i+r}), \dots), \quad (1)$$

using periodic boundary conditions, *i.e.* for any $i \in \mathbb{Z}$ it holds that $s_{i+N} = s_i$.

The function f_A used in this definition will be referred to as a local rule, and the integer r will be referred to as the radius of the neighborhood. Any local rule can be uniquely defined by a lookup table (LUT) that lists all of the possible arguments together with the corresponding function values. It is assumed that the arguments are listed in a lexicographic order. The general form of such a LUT in the case of radius $r = 1$ is shown in Table I.

TABLE I. LUT OF LOCAL RULE $R = (l_8, l_7, l_6, l_5, l_4, l_3, l_2, l_1)_2$

111	110	101	100	011	010	001	000
l_8	l_7	l_6	l_5	l_4	l_3	l_2	l_1

The LUT can be used to enumerate local rules, as the coefficients l_i can be treated as digits in the binary representation of an integer R , *i.e.* $R = \sum_{i=1}^8 l_i 2^{i-1}$. Clearly this extends to higher values of the radius. Due to the fact that the ordering of arguments in the LUT is fixed, only the second row needs to be stored, such that a LUT may be represented as a binary vector. The length of such a vector is 2^{2r+1} .

With $\{0, 1\}^*$ we will denote the set of all binary sequences of finite length, *i.e.* $\{0, 1\}^* = \bigcup_{N \geq 0} \{0, 1\}^N$. The function $A : \{0, 1\}^* \rightarrow \{0, 1\}^*$, satisfying $A(\bar{X}) = A_N(X)$ if and only if $X \in \{0, 1\}^N$, where each of the global rules A_N is defined with the same local rule f_A , will be referred to as a generalized global rule of a CA. Such functions will be frequently used throughout this paper, therefore we will simply refer to them as global rules or rules. In this paper a CA will be identified in terms of its global rule, and by referring to a CA we therefore always refer to its global rule in this generalized sense.

Note that rule A is uniquely defined by a given local rule f_A , but the opposite is not true. For a given rule A we may find different local rules defining it. Fact 1 highlights the relationship between different local rules defining the same CA.

Fact 1. Two local rules $f: \{0,1\}^{2r+1} \rightarrow \{0,1\}$ and $g: \{0,1\}^{2u+1} \rightarrow \{0,1\}$, $u \leq r$, define the same CA if and only if it holds:

$$f(s_1, \dots, s_{2r+1}) = g(s_{r-u+1}, \dots, s_{r+u+1}), \quad (2)$$

for any $(s_1, \dots, s_{2r+1}) \in \{0,1\}^{2r+1}$.

Example 1. Let $g: \{0,1\} \rightarrow \{0,1\}$ be defined by $g(s) = s$ and $f: \{0,1\}^3 \rightarrow \{0,1\}$ be defined by $f(s_1, s_2, s_3) = s_2 s_3 + s_2(1 - s_3)$. We can see that for any $s_1, s_3 \in \{0,1\}$ it holds that $f(s_1, s_2, s_3) = g(s_2) = s_2$, and thus f and g define the same CA rule, which happens to be the identity rule. \square

For a given neighborhood radius r , \mathcal{A}_r denotes the set of all CAs that can be expressed with the use of a local rule with a neighborhood of radius r . CAs belonging to \mathcal{A}_1 are referred to as Elementary CAs (ECAs), and form the most commonly studied class of 2-state CAs [19].

Two important properties of the sets \mathcal{A}_r are underlined in Fact 2.

Fact 2. For any $r \geq 0$, $\mathcal{A}_r \subset \mathcal{A}_{r+1}$ and $|\mathcal{A}_r| = 2^{2^{2r+1}}$.

Let A be a CA, $X \in \{0,1\}^M$ for some M and $T > 0$. The finite sequence of vectors given by:

$$(X, A(X), A^2(X), \dots, A^{T-1}(X)),$$

where A^t denotes the t -th application of the rule A , will be referred to as the space-time diagram containing T time steps. Each of the elements of a space-time diagram will be referred to as a configuration of the CA, while the first element will be referred to as the initial configuration. If $t = 0, 1, \dots, T-1$ and $m = 1, \dots, M$, then $A^t(X)[m]$ refers to the state of the m -th cell in the t -th element of the space-time diagram.

Example 2. We consider an ECA defined by local rule 150. The LUT of ECA 150 is shown in Table II.

TABLE II. LUT OF ECA 150

111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

Figure 1 depicts a space-time diagram of ECA 150, starting from a random initial configuration. Following a common convention, the space-time diagram is visualized as a bitmap in which every row corresponds to a configuration at specific time step. The first row in the image is the initial configuration. State one is drawn as a black pixel, while white pixel corresponds to state zero. \square

III. PROBLEM STATEMENT

In this section we define the identification problem. The formulation presented below is based on the concept of an observation of a space-time diagram, which is assumed to be incomplete, *i.e.* it contains only partial information on the states of the CA.

Formally, we assume that the states of a system, which is assumed to be an unknown CA, were observed at certain, unknown time steps. Let I be an $N \times M$ array containing symbols belonging to the set $\{0,1,?\}$, where the symbols 0 and 1 denote valid states, while ? denotes an unknown state

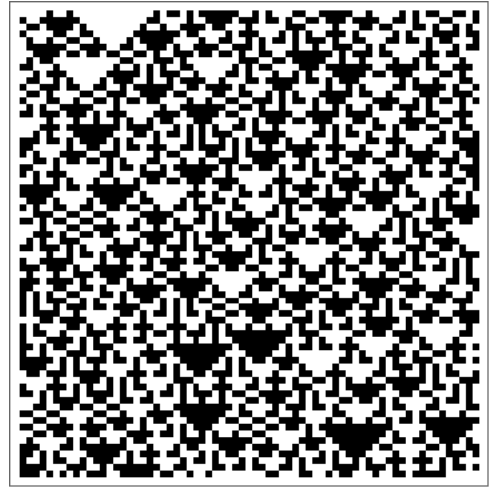


Fig. 1. Space-time diagram of ECA 150

belonging to the set $\{0,1\}$. Additionally, let the first row $I[1] \in \{0,1\}^M$. Such an array I will be referred to as an observation. If an observation I does not contain the symbol ?, we refer to it as spatially complete. The first row $I[1]$ is assumed to represent the initial configuration of a CA, and row $I[n]$ for $n > 1$ represents the configuration at time step τ_n . It is assumed that $\tau_n < \tau_{n+1}$.

Let I be an observation. The number $C(I) = \#\{I[n, m] \neq ?\}$ will be referred to as the number of completely observed states. In our case, for any observation I it holds that $C(I) > 0$.

For each observation I , we define the set $\text{com}(I)$ that contains all of the spatially complete observations I' , satisfying $I'[n, m] = I[n, m]$ for all n, m such that $I[n, m] \neq ?$.

Example 3. Let observation I be given by:

$$I = \begin{bmatrix} 0 & 1 & 0 \\ 0 & ? & 1 \\ 1 & 1 & ? \end{bmatrix}.$$

Then the set $\text{com}(I)$ is given by:

$$\text{com}(I) = \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right\}.$$

As can be easily counted, $C(I) = 7$. \square

We will say that a CA A fits the observation I if and only if there exists an $I' \in \text{com}(I)$ and a sequence of natural numbers (τ_n) such that $\tau_n < \tau_{n+1}$ and for any $n \in \{1, 2, \dots, N-1\}$ it holds:

$$A^{\tau_n}(I'[1]) = I'[n+1]. \quad (3)$$

Proposition 3. Rule A fits the observation I if and only if there exist an $I' \in \text{com}(I)$ and a sequence of natural numbers (t_n) such that $t_n \leq t_{n+1}$ and for any $n \in \{1, 2, \dots, N-1\}$ it holds:

$$A^{t_n}(I'[n]) = I'[n+1]. \quad (4)$$

The sequence (τ_n) in the definition of fitting, corresponds to the time steps in the CA evolution (which are assigned to the rows of the observation), while the sequence (t_n) in Proposition 3 refers to the number of missing time frames between two consecutive rows in the observed diagram. Obviously $\tau_n = \sum_{i=1}^n t_i$.

In practice, it is useful to be able to use more than one observation for the identification. Therefore, we will consider observation sets \mathcal{I} containing a finite number of observations. For simplicity, we assume that the elements of \mathcal{I} are numbered, i.e. $\mathcal{I} = \{I_1, \dots, I_{|\mathcal{I}|}\}$. We will say that rule A fits the observation set \mathcal{I} , if it fits all of the observations in the set.

Note that for the sake of simplicity we will write $C(\mathcal{I})$ to express the number of observed states in all of the observations belonging to \mathcal{I} , i.e. $C(\mathcal{I}) = \sum_{I \in \mathcal{I}} C(I)$. Additionally, we will write $M(\mathcal{I})$ to denote the total number of columns in all of the observations belonging to \mathcal{I} , i.e. $M(\mathcal{I}) = \sum_{I \in \mathcal{I}} M_I$ where M_I is the number of columns of observation I .

For a non-empty observation set \mathcal{I} , the set $\mathcal{R}(\mathcal{I})$ will denote all CA rules that fit the observation set \mathcal{I} . The identification problem is defined as finding the elements of the set $\mathcal{R}(\mathcal{I})$ based on \mathcal{I} . In practice, our goal will be limited to finding at least one of the elements of $\mathcal{R}(\mathcal{I}) \cap \mathcal{A}_r$ for some $r > 0$. The problem can also be seen from the machine learning perspective in which the observation set is a training set, from which we try to learn and build a set of rules based on this knowledge.

The following fact will be used in the design of the identification algorithm, to simplify calculations. Informally, it could be expressed by understanding the observation set \mathcal{I} as a set of conditions that the rule needs to meet. Having fewer conditions, it becomes more likely to find solutions meeting those conditions.

Fact 4. Let \mathcal{I} be an observation set, and let $\mathcal{I}' \subset \mathcal{I}$. Then $\mathcal{R}(\mathcal{I}) \subset \mathcal{R}(\mathcal{I}')$.

Since we consider only finite observation sets, we know that for every observation set \mathcal{I} there exists a $T > 0$ such that, if a solution exists, and (t_n^I) is the time gap sequence of observation $I \in \mathcal{I}$, then $1 \leq t_n^I \leq T$, for every n . In the construction of the solution algorithm, we will assume that an upper-bound for T is known.

IV. CA IDENTIFICATION AS AN OPTIMIZATION PROBLEM

The identification problem, defined in Section III, can be formulated as an optimization problem, which in turn enables the use of evolutionary search methods.

We start with an auxiliary definition. Let $a, b \in \{0, 1, ?\}^M$ be some vectors. We define the distance between a and b as:

$$\text{dist}(a, b) = \sum_{a_i, b_i \in \{0, 1\}} |a_i - b_i|. \quad (5)$$

We assume that if there is no i such that $a_i \neq ?$ and $b_i \neq ?$ then $\text{dist}(a, b) = 0$. Therefore $\text{dist}(a, b) = 0 \nRightarrow a = b$.

Assume that \mathcal{I} is a set of observations of some unknown CA belonging to \mathcal{A}_r , i.e. $\mathcal{R}(\mathcal{I}) \cup \mathcal{A}_r \neq \emptyset$. Let A be a CA, and

for every $I \in \mathcal{I}$, let (τ_n^I) be a strictly increasing sequence of natural numbers.

As a start, we define the error measure $E_{\mathcal{I}}(A, (\tau_i^I))$, which measures how well a given CA A fits the observation set \mathcal{I} , assuming that τ_i^I is the time step of the i -th row in observation I . The measure $E_{\mathcal{I}}$ is defined as:

$$E_{\mathcal{I}}(A, (\tau_i^I)) = \sum_{I \in \mathcal{I}} \sum_{n=1}^{N_I-1} \text{dist}(A^{\tau_n^I}(I[1]), I[n+1]), \quad (6)$$

where N_I is the number of rows of observation $I \in \mathcal{I}$. The following fact is a direct consequence of the definition of the identification problem.

Fact 5. $A \in \mathcal{R}(\mathcal{I})$ if and only if there exists a sequence (τ_i^I) such that $E_{\mathcal{I}}(A, (\tau_i^I)) = 0$.

Note that in the case when $\mathcal{I} = \{I\}$ we will write E_I instead of $E_{\{I\}}$.

Let (t_i) be a sequence of natural numbers, and let A be a CA rule. Observation $\bar{I}_{(t_i)}^A$ defined as:

$$\bar{I}_{(t_i)}^A[n, m] = \begin{cases} I[n, m], & \text{if } I[n, m] \neq ?, \\ A^{t_{n-1}}(\bar{I}_{(t_i)}^A[n-1])[m], & \text{if } I[n, m] = ?, \end{cases}$$

will be referred to as the A -completion of I with time gaps (t_i) . Note that any observation I satisfies $I[1] = \bar{I}_{(t_i)}^A[1]$ for any $A, (t_i)$.

Fact 6. $\bar{I}_{(t_i)}^A \in \text{com}(I)$.

Example 4. Assume that CA A is ECA 150 with LUT given by Table II and local rule f_{150} . Let $(t_i)_{i=1}^2 = (1, 2)$. We consider the observation I defined in Example 3 and compute $\bar{I}_{(t_i)}^A$.

$$I = \begin{bmatrix} 0 & 1 & 0 \\ 0 & ? & 1 \\ 1 & 1 & ? \end{bmatrix} \quad \bar{I}_{(t_i)}^A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The calculation is as follows. Firstly we compute $\bar{I}_{(t_i)}^A[2, 2]$. Since $t_1 = 1$ we simply apply the rule to the first row of I , i.e. $\bar{I}_{(t_i)}^A[2, 2] = f_{150}(I[1, 1], I[1, 2], I[1, 3]) = 1$. Since $t_2 = 2$, to find $\bar{I}_{(t_i)}^A[3, 3]$, we first need to compute one additional configuration by evaluating the rule on configuration $\bar{I}_{(t_i)}^A[2]$. It is easy to check that $A(\bar{I}_{(t_i)}^A[2]) = (0, 0, 0)$, and thus $\bar{I}_{(t_i)}^A[3, 3] = f_{150}(0, 0, 0) = 0$. \square

Based on Proposition 3, we define an alternative error measure $\tilde{E}_{\mathcal{I}}(A, (t_i^I))$ that will turn out to be more useful in the construction of the solution algorithm. Assuming that (t_i^I) is a sequence of natural numbers representing time gaps, $\tilde{E}_{\mathcal{I}}$ is defined as:

$$\tilde{E}_{\mathcal{I}}(A, (t_i^I)) = \sum_{I \in \mathcal{I}} \sum_{n=1}^{N_I-1} \text{dist}\left(A^{t_n^I}(\bar{I}_{(t_i^I)}^A[n]), \bar{I}_{(t_i^I)}^A[n+1]\right). \quad (7)$$

Since $\bar{I}_{(t_i)}^A \in \text{com}(I)$, we can express $\tilde{E}_{\mathcal{I}}$ without using the function dist as:

$$\tilde{E}_{\mathcal{I}}(A, (t_i^I)) = \sum_{I \in \mathcal{I}} \sum_{n=1}^{N_I-1} |A^{t_n^I}(\bar{I}_{(t_i^I)}^A[n]) - \bar{I}_{(t_i^I)}^A[n+1]|. \quad (8)$$

Example 5. We refer again to observation I , CA A and (t_i) used in Example 4 and we compute the error measures E_I and \tilde{E}_I . Let us start with E_I . Following the fact that $\tau_n = \sum_{i=1}^n t_i$, we get $(\tau_i) = (1, 3)$. The error measure E_I can be computed easily by evolving A , starting from the initial configuration $I[1]$ and comparing the results with the values in I , for entries not occupied by $?$.

Starting from the top: $A(I[1]) = (1, 1, 1)$. Since $\tau_1 = 1$ we compare the outcome with the second row of I . As we see, $I[2, 1] = 0 \neq 1$ has an incorrect value, $I[2, 2] = ?$ so it does not contribute to the error and $I[2, 3] = 1$ which is a correct value. Since $\tau_2 = 3$ we should further evolve A three times, starting from $A(I[1])$, but since $A((1, 1, 1)) = (1, 1, 1)$, we can simply compare $I[3]$ with $(1, 1, 1)$ and see that no errors occur. Summing up, the total error is: $E_I(A, (\tau_i)) = 1$.

Similarly, we find the value of \tilde{E}_I , by taking pairs of rows $\bar{I}_{(t_i)}^A[n]$ and $I[n+1]$ and comparing the results of $A^{t_n}(\bar{I}_{(t_i)}^A[n])$ and $I[n+1]$. The error in the first pair of rows is the same as in the case of E_I . For the second pair the initial condition is $\bar{I}_{(t_i)}^A[2] = (0, 1, 1)$, and since $A(0, 1, 1) = (0, 0, 0)$ and since $A((0, 0, 0)) = (0, 0, 0)$, we do not further evaluate A . We compare $(0, 0, 0)$ with $I[3]$, which yields 2 incorrect values. Summing up, the total error is $\tilde{E}_I(A, (t_i)) = 3$. \square

The relation between $E_{\mathcal{I}}$ and $\tilde{E}_{\mathcal{I}}$ is expressed by the following proposition.

Proposition 7. *Let A be a CA rule and \mathcal{I} an observation set. There exists a strictly increasing sequence (τ_i^I) of natural numbers, such that $E_{\mathcal{I}}(A, (\tau_i^I)) = 0$ if and only if there exists a sequence (t_i^I) of natural numbers such that $\tilde{E}_{\mathcal{I}}(A, (t_i^I)) = 0$.*

As a consequence of Proposition 7, the identification problem can be expressed mathematically as the minimization of \tilde{E} . Note that this is only possible due to the assumption that observation set \mathcal{I} contains partial space-time diagrams of some unknown CA. In a more general setting, where the observations could have a more complex origin, such a simplification is not possible.

As mentioned earlier, we consider the case where the upper bound for the time gaps is known. Using this knowledge, we define the error measure $\tilde{E}_{\mathcal{I}}$ independently of the selection of (t_i^I) as:

$$\tilde{E}_{\mathcal{I}}(A) = \min_{\substack{(t_i^I) \\ 1 \leq t_i^I \leq T}} \tilde{E}_{\mathcal{I}}(A, (t_i^I)). \quad (9)$$

Note that the minimum in (9) is always defined, since there is a finite number of possibilities for the choice of t_n^I . Additionally, note that for a spatially complete observation I , the choice of t_n^I is independent of the choice of t_m^I for any $n \neq m$, and for observations I and J , the choice of (t_i^I) is independent from the choice of (t_i^J) . Consequently, to find the value of $\tilde{E}_{\mathcal{I}}$ in the case of a spatially complete observation set, we need to examine at most $\sum_{I \in \mathcal{I}} T(N_I - 1)$ sequences of time gap lengths.

In the general case, the choices of the values of (t_i^I) are dependent on each other, and thus in order to find the exact value of the error measure we need to examine all of the

T^{N_I-1} possibilities, which holds a substantial computational burden. Due to this, even in the case of partial observations, we follow the approach described above and treat the time steps independently. The only difference that we introduce is that if for given n , few different candidate values for t_n^I lead to the same, minimal value of the pairwise error, one of those candidates is being selected randomly. Such an approach, is a stochastic overestimation of the error, *i.e.* the calculated value will never be lower than the actual error. Additionally, if a given CA is a solution to the problem, recalculating the approximate error measure multiple times increases the probability of finding the exact value, which is found by taking the minimum of all of the obtained results. Such an approach turned out to be sufficient in the discussed context.

V. EVOLUTIONARY ALGORITHM

Having stated the identification problem as an optimization problem in this section, we describe its solution using an evolutionary algorithm based on the classical GA. In order to follow the GA approach, we need to define the individuals' representation, the population structure, a fitness function for ranking the individuals, but also the selection procedure for reproduction, and finally the cross-over and mutation operators. Formally, also halting conditions need to be formulated.

A. Representation of individuals and population structure

Here, the individuals that make up the population are CAs, encoded through the LUT of their local rules, which is possible since the LUT of any CA $A \in \mathcal{A}_r$ can be represented as a bit-string of length $2^{2^{r+1}}$. We assume that the population consists of CA belonging to \mathcal{A}_r , for some $r > 0$.

We consider populations of $P > 0$ individuals. By \mathcal{P}^i we denote the population of the i -th generation of the GA. The population \mathcal{P}^1 is the initial population, and is constructed by randomly selecting P bit-strings. Populations \mathcal{P}^i for $i > 1$ are the outcomes of applying the genetic operators, according to the rules described in the remainder of this section.

B. Fitness function

The fitness function is directly related to the error measure $\tilde{E}_{\mathcal{I}}$ defined by (9). Although Proposition 7 states that the error measures given by (6) and (9) can be used interchangeably, preliminary experiments showed that the later results in efficient and convergent algorithm, while suboptimal results were obtained using the measure given by (6). This follows from the fact that the error in row n is affected by errors appearing in rows $2, \dots, n-1$. As we know from the research on dynamical properties of CAs, small initial perturbations can strongly affect the final system state [20]. For that reason, it is easier to optimize $\tilde{E}_{\mathcal{I}}$ with a GA as compared to $E_{\mathcal{I}}$.

Let $L \in \{0, 1\}^{2^{2^{r+1}}}$ be a LUT of some local rule which defines a CA A . Then $\text{fit}_{\mathcal{I}}(L)$ denotes the fitness of A , and is defined as:

$$\text{fit}_{\mathcal{I}}(L) = C(\mathcal{I}) - M(\mathcal{I}) - \tilde{E}_{\mathcal{I}}(A). \quad (10)$$

The fitness function takes integer values from 0 up to $C(\mathcal{I}) - M(\mathcal{I})$, *i.e.* there are finitely many possible values of the fitness function. The goal of the GA is to maximize fitness, and a CA

with a maximal fitness value is a solution of the identification problem. From the above, it is clear that if $C(\mathcal{I}) - M(\mathcal{I})$ is close to zero, solving the problem is infeasible, since the number of possible values is very small and the population is not able to gradually increase its fitness. Additionally, if $C(\mathcal{I}) = M(\mathcal{I})$, then the problem is trivial because every CA is a solution.

The fitness defined by (10) has proven to work effectively, but the computing time needed for its evolution becomes unacceptable if the observation set is large. Therefore, during the evolution, to estimate the value of $\text{fit}_{\mathcal{I}}$ we use $\text{fit}_{\mathcal{I}'}$ for some non-empty subset $\mathcal{I}' \subset \mathcal{I}$. We start by randomly selecting elements for the subset \mathcal{I}' . Subsequently, but before evolving a new population we replace one of the elements in the subset \mathcal{I}' with a randomly selected observation from \mathcal{I} . Due to Fact 4 we are sure that such an approach does not result in reducing the solution set.

C. Selection operator

Having defined the fitness function, we can define the selection operator, which is responsible for selecting the parent individuals that will be used to produce the next generation. We use a random selection method where the selection probability of a given individual is proportional to its fitness. Individuals are selected with replacement, *i.e.* individuals might be selected multiple times for reproduction.

D. Cross-over operator

To produce offspring, we select two parents according to the procedure described in Subsection V-C. A uniform cross-over operator is used, *i.e.* if L_1, L_2 denote parents, the outcome of the cross-over operator is a vector L_c with values that are randomly selected from L_1 and L_2 , *i.e.* $\mathbb{P}(L_c[i] = L_1[i]) = \mathbb{P}(L_c[i] = L_2[i]) = 0.5$.

E. Mutation operator

Finally, the offspring individual is mutated. A simple bit-flip mutation is being used, *i.e.* for every position of the vector a decision is made whether or not the value should be flipped, with p_f being the probability of flipping the value. The expected number of flipped positions in the population is $P p_f 2^{2r+1}$.

F. Elite survival

After evolving a new population, the elite survival procedure is applied. Our experiments proved that such an approach is required to reach convergence. The procedure is implemented by a deterministic selection of $P_E \ll P$ fittest individuals from the previous population used to replace randomly selected individuals in the newly evolved one.

Including this elite survival process can dramatically increase the performance of the algorithm, though there are cases where such an approach causes the population to progress towards a local optimum. To overcome this, we apply a simple, adaptive procedure that deactivates elite survival in cases when the maximum fitness value of the population remained constant for more than N_{off} generations. The elite survival procedure is again switched on after a predefined number of N_{on} generations, or if the maximum fitness improved.

G. Halting conditions

The algorithm evolves by generating populations according to the procedure described above until a maximum, predefined number of populations Λ was evolved or, if a CA that fits the observation set was discovered.

As mentioned in Subsection V-B during the evolution, the fitness $\text{fit}_{\mathcal{I}}$ is approximated by $\text{fit}_{\mathcal{I}'}$ for some $\mathcal{I}' \subsetneq \mathcal{I}$, which is effective for selection, but can not be used in the halting condition since $\text{fit}_{\mathcal{I}'}(A) = C(\mathcal{I}') - M(\mathcal{I}')$ does not imply $\text{fit}_{\mathcal{I}}(A) = C(\mathcal{I}) - M(\mathcal{I})$. Therefore, for the individual A with the highest value $\text{fit}_{\mathcal{I}'}(A)$, we additionally calculate $\text{fit}_{\mathcal{I}}(A)$ and base the halting condition on it, *i.e.* the algorithm stops as soon an element is found.

VI. RESULTS OF EXPERIMENTS

By means of our experiments we verified to what extent the partiality of observations affects the efficiency of the GA in terms of the number of GA iterations required to find a solution.

We concentrated on two ECAs: 150 and 180, with LUTs given in Table II and III, respectively.

TABLE III. LUT of ECA 180

111	110	101	100	011	010	001	000
1	0	1	0	1	0	1	0

In this experiment, the GA evolution is based on observation sets $\mathcal{I}_A(k)$ for $k = \{0, 1, \dots, 150\}$ and ECA $A \in \{150, 180\}$. The integer k will be referred to as the problem number. The observation set $\mathcal{I}_A(0)$ is a set of $\Omega > 0$ observations obtained from Ω different, random initial conditions common for both A , by selecting subsequent configurations of ECA A generating time gaps of random length from 1 to T . The set $\mathcal{I}_A(k)$ for $k > 0$ is built from observations belonging to $\mathcal{I}_A(k-1)$ by modifying them in such a way that $\pi = 2000$ randomly selected, completely observed entries are replaced by “?”. In other words, by increasing k the effect of spatial partiality is increased. As a result of such a procedure we obtained a series of observation sets $(\mathcal{I}_A(k))_{k=0}^{150}$ for which it holds $C(\mathcal{I}_A(k)) - C(\mathcal{I}_A(k+1)) = \pi$. The identification algorithm was then executed for each of the obtained observation sets.

Given that the family of ECAs contains only 256 members, the identification problem would be relatively easy to tackle, so we set the radius $r = 2$, *i.e.* the population contains local rules with radius $r = 2$ represented as bit-strings of length 32. Without this modification the algorithm is able to find a solution in a few iterations, by examining the entire search space.

In order to account for the stochastic nature of the GA, the experiment is repeated $L > 0$ times for each r, k . The values of the GA parameters used in our experiment setup are shown in Table IV.

The results vary significantly depending on the rule in question, which is not surprising since the dynamics of ECAs 150 and 180 is different. The normalized Maximum Lyapunov Exponent (nMLE) [21]–[23] of the former is the highest among

TABLE IV. PARAMETERS USED IN THE EXPERIMENT

param	value	description
r	2	rule radius
p_f	0.01	probability of flipping 1-bit in mutation
P	512	number of individuals in population
P_E	32	elite size
T	10	bound for the time gap length
C	69	number of rows / columns in each observation
Ω	64	number of observations
π	2000	number of cells being removed from each observation set
s	8	number of samples for fitness approximation
Λ	5000	maximal number of the GA populations
L	20	number of repetitions of the GA per rule

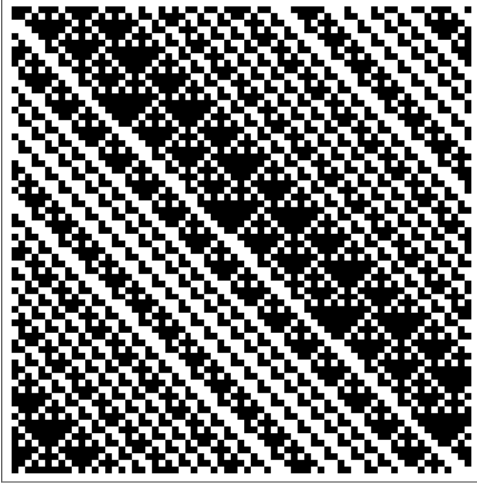


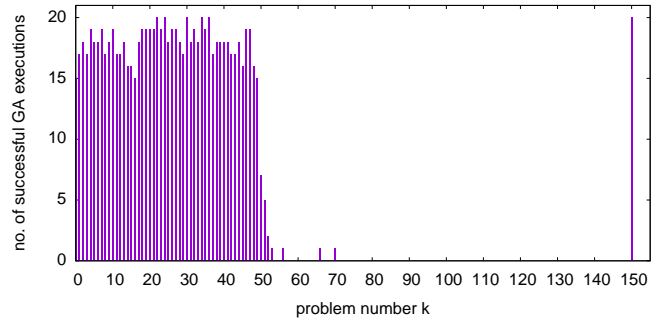
Fig. 2. Space-time diagram of ECA 180

all of the ECAs, and thereby this CA's behavior may be considered complex. In contrast, the nMLE of ECA 180 is only approximately 0.48, which hints that, in some sense, the behavior of this ECA is simpler than the one displayed by ECA 150. The differences in the overall dynamical complexity of these two CAs can be acknowledged by examining their space-time diagrams, which are depicted in Fig. 1 and Fig. 2.

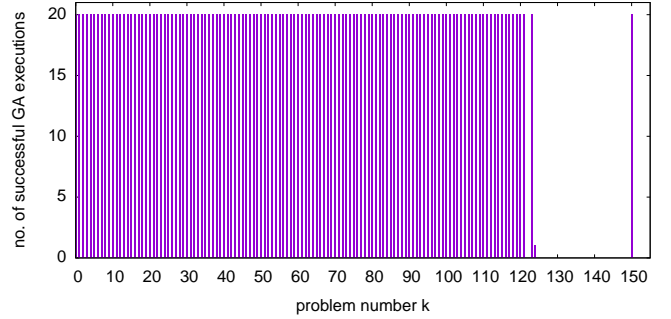
To understand the performance of the GA, we first checked for which k the algorithm was able to find a solution (Fig. 3). When comparing the plot for ECA 150 with the one for ECA 180, it is clear that the identification problem turned out to be much more challenging for ECA 150. Indeed, for this ECA, the algorithm was effective only if the number removed observation elements was smaller than 50π , whereas it mostly failed when more spatial partiality was added. Besides, even for k close to 0, not all of the GA executions were successful. In contrast, identifying ECA 180 was always possible for $k < 120$, but for $k > 120$ we see a sudden drop in the performance. Note that in both cases, for $k = 150$ a solution was easily found, since for this setting the problem is trivial, *i.e.* almost all CAs can be considered a solution.

The above results suggest that, depending on the dynamical characteristics of the CA in question, the maximum allowable number of missing elements in the observations differs. Further research is undertaken to better understand the link between the identifiability and dynamics of CAs.

Figure 4 depict the minimum, average and maximum number of GA iterations among the runs resulting in a solution for ECA 150 and ECA 180, respectively. In the case of ECA



(a) ECA 150



(b) ECA 180

Fig. 3. Number of successful GA executions

180, we see that the efforts needed for finding a solution grows as k increases, up to the point where it becomes impossible. Furthermore, we see that in most cases the difference between maximal and minimal values is relatively low. In the case of ECA 150, the results are much less stable. The differences between maximal and minimal values are substantial, and the efforts needed to find the solution do not steadily grows with the growing spatial partiality. The only similarity between the two CAs seems to be in the fact that there exists some critical k beyond which the problem becomes impossible to solve.

SUMMARY

In this paper we introduced the identification problem of CAs in the context of partial observations. An evolutionary algorithm for tackling the problem was presented, and its performance was verified for the two ECAs. The initial experiments suggest that the difficulty of the identification problem is somehow linked to the dynamical complexity of the CAs. The problem and solution algorithm presented in this paper, should be considered as one of the first steps in identifying CAs from data originating from real-world phenomenon observations. Unavoidably, such observations will be somehow incomplete in the sense that it is impossible to continuously track the involved processes.

REFERENCES

- [1] S. Al-Kheder, J. Wang, and J. Shan, "Cellular automata urban growth model calibration with genetic algorithms," in *Proc. Urban Remote Sensing Joint Event, 2007*. IEEE, 2007, pp. 1–5.
- [2] E. Sapin, L. Bull, and A. Adamatzky, "Genetic approaches to search for computing patterns in cellular automata," *IEEE Comput. Intell. Mag.*, vol. 4, no. 3, pp. 20–28, 2009.

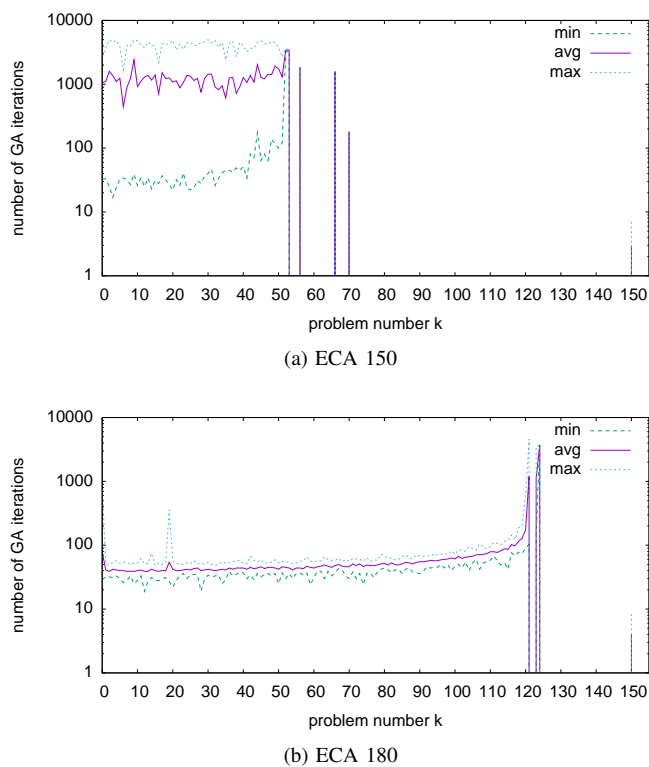


Fig. 4. Number of GA iterations required to find a solution

[3] P. L. Rosin, "Image processing using 3-state cellular automata," *Comput. Vis. Image Underst.*, vol. 114, no. 7, pp. 790–802, 2010.

[4] F. C. Richards, T. P. Meyer, and N. H. Packard, "Extracting cellular automaton rules directly from experimental data," *Physica D: Nonlinear Phenomena*, vol. 45, no. 1, pp. 189–202, 1990.

[5] M. Mitchell, J. P. Crutchfield, and R. Das, "Evolving cellular automata with genetic algorithms: A review of recent work," in *Proceedings of the First International Conference on Evolutionary Computation and its Applications (EvCA'96)*, 1996.

[6] T. Bäck, R. Breukelaar, and L. Willmes, "Inverse Design of Cellular Automata by Genetic Algorithms: An Unconventional Programming Paradigm," in *Unconventional Programming Paradigms*, ser. Lecture Notes in Computer Science, J.-P. Banâtre, P. Fradet, J.-L. Giavitto, and O. Michel, Eds. Springer-Verlag, 2005, vol. 3566, pp. 161–172.

[7] E. Sapin, O. Bailleux, and J.-J. Chabrier, "Research of a cellular automaton simulating logic gates by evolutionary algorithms," in *Proceedings of the 6th European conference on Genetic programming*, ser. EuroGP'03. Springer-Verlag, 2003, pp. 414–423.

[8] S. Bandini, S. Manzoni, and L. Vanneschi, "Evolving robust cellular automata rules with genetic programming," in *Automata*, A. Adamatzky, R. Alonso-Sanz, A. T. Lawniczak, G. J. Martínez, K. Morita, and T. Worsch, Eds. Luniver Press, Frome, UK, 2008, pp. 542–556.

[9] K. Maeda and C. Sakama, "Identifying cellular automata rules," *J. Cellular Automata*, vol. 2, no. 1, pp. 1–20, 2007.

[10] D. Andre, F. H. Bennett III, and J. R. Koza, "Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem," in *Proceedings of the First Annual Conference on Genetic Programming*. MIT Press, 1996, pp. 3–11.

[11] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[12] X. Liu, X. Li, L. Liu, J. He, and B. Ai, "A bottom-up approach to discover transition rules of cellular automata using ant intelligence," *Int. J. Geogr. Inf. Sci.*, vol. 22, no. 11-12, pp. 1247–1269, 2008.

[13] L. Bull and A. Adamatzky, "A learning classifier system approach to the identification of cellular automata," *J. Cellular Automata*, vol. 2, no. 1, pp. 21–38, 2007.

[14] A. Adamatzky, *Identification Of Cellular Automata*. Taylor & Francis Group, 1994.

[15] Y. Yang and S. A. Billings, "Extracting Boolean rules from CA patterns," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, no. 4, pp. 573–580, 2000.

[16] —, "Neighborhood detection and rule selection from cellular automata patterns," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, no. 6, pp. 840–847, 2000.

[17] X. Sun, P. L. Rosin, and R. R. Martin, "Fast rule identification and neighborhood selection for cellular automata," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 3, pp. 749–760, 2011.

[18] W. Bolt, J. M. Baetens, and B. De Baets, "Identifying CAs with evolutionary algorithms," in *Proceedings 19th International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA 2013) – Exploratory Papers*, September 2013, pp. 11–20.

[19] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, pp. 601–644, Jul 1983.

[20] J. M. Baetens and B. De Baets, "Phenomenological study of irregular cellular automata based on Lyapunov exponents and Jacobians," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, 2010.

[21] S. Wolfram, "Universality and complexity in cellular automata," *Physica D*, vol. 10, pp. 1–35, 1984.

[22] M. Shereshevsky, "Lyapunov exponents for one-dimensional cellular automata," *Journal of Nonlinear Science*, vol. 2, no. 1, pp. 1–8, 1992.

[23] F. Bagnoli, R. Rechtman, and S. Ruffo, "Damage spreading and Lyapunov exponents in cellular automata," *Physics Letters A*, vol. 172, no. 1-2, pp. 34–38, Dec. 1992.